K-Spectral Centroid : Extension and Optimizations

Brieuc Conan-Guez, Alain Gély, Lydia Boudjeloud-Assala, Alexandre Blansché

Université de Lorraine, CNRS, LORIA, F-57000 Metz, France
{brieuc.conan-guez, alain.gely, lydia.boudjeloud-assala, alexandre.blansche}
@univ-lorraine.fr

Abstract. In this work, we address the problem of unsupervised classification of large time series datasets. We focus on K-Spectral Centroid (KSC), a k-means-like model, devised for time series clustering. KSC relies on a custom dissimilarity measure between time series, which is invariant to time shifting and Y-scaling. KSC has two downsides: firstly its dissimilarity measure only makes sense for non negative time series. Secondly the KSC algorithm is relatively demanding in terms of computation time. In this paper, we present a natural extension of the KSC dissimilarity measure to time series of arbitrary signs. We show that this new measure is a metric distance. We propose to speed up this extended KSC (EKSC) thanks to four exact optimizations. Finally, we compare EKSC to a similar model, K-Shape, on real world datasets.

1 Extension of K-Spectral Centroid

1.1 Introduction

Unsupervised classification of time series has been receiving a great deal of attention for many years. In this work, we focus on a specific model, K-Spectral Centroid (KSC)[1], a K-means-like algorithm devised for time series clustering. KSC distinguishes itself from other classic partitioning models by relying on a specific shape-based dissimilarity measure: this measure is invariant to Y-scaling and global time shifting (contrary to the classic DTW measure which applies a non uniform transformation of the time axis). Invariance to scaling (magnitude in Y values) is classically addressed through a Y-normalization, whereas invariance to shift-lag is obtained by testing all possible time translations of one time series with respect to the other time series. This shape based measure is well adapted to time series for which it is difficult to define a time origin.

Even if KSC has been succesfully applied to several real world problems (e.g. the temporal evolution of hashtags on Twitter [1]), this model suffers from two downsides: firstly its dissimilarity measure can only handle non negative time series. Secondly, KSC has a running time complexity which is cubic in the time series length. Processing massive data is therefore very time consuming.

In this work, we address these two downsides. We show that KSC can naturally be extended to cope with time series of arbitrary signs by generalizing its dissimilarity measure. We denote EKSC this extension of KSC. Moreover, we prove that the new dissimilarity measure is a metric distance (the triangle inequality). In the second part of this work, we reduce the complexity and the

running time of EKSC thanks to four exact optimizations. In particular, we apply Elkan's algorithm [2] which relies on the triangle inequality to cut off a lot of distance evaluations.

In the experimental part, we focus on evaluating the solution quality of EKSC. K-Shape [3], which is a similar model, is used as comparison. Finally, EKSC accelerations obtained thanks to the four optimizations are reported.

1.2 K-Spectral Centroid for arbitrary time series

Let x and y be two discrete one-dimensional time series. Time series lengths, $L(x)$ and $L(y)$, can be different. Adding two time series is carried out thanks to a suitable null padding. We recall the expression of d_+, the shape based dissimilarity measure used in KSC to compare non negative time series:

$$d_+(x,y) = \min_{o\in\mathcal{Z},\alpha\in R} \frac{\|x - \alpha\ \tau_o(y)\|}{\|x\|}$$

$\|.\|$ is the Euclidean norm, and τ_o is the time shift operator of parameter o (a signed integer). Minimization with respect to α (a real number) addresses the problem of magnitude scale, whereas minimization with respect to o provides a time series alignment which is time-shift invariant.

In [1], Yang and Leskovec show that this dissimilarity measure is symmetric thanks to a simple reformulation of d_+. If we denote $g(c) = \sqrt{1-c^2}$ and $x \cdot y$ the scalar product (with null padding), we have $d_+(x,y) = \min_{o\in\mathcal{Z}} g\left(\frac{x\cdot\tau_o(y)}{\|x\|\|y\|}\right)$.

The measure d_+ is only defined for non-negative time series (the notation uses subscript $+$ for this purpose). Indeed, d_+ collates a time series x and its opposite ($\alpha = -1$ as α is not constraint to be non negative), which makes no sense in many real case applications. In this work, we propose to define a new measure d, an extension of d^+ to time series of arbitrary signs. We denote $d(x,y) = g\left(\frac{\max_{o\in\mathcal{Z}}(x\cdot\tau_o(y))}{\|x\|\|y\|}\right)$. d is symmetric, and we have $d_+(x,y) \leq d(x,y)$. We have an equality for non negative time series.

We show now that d verifies the triangle inequality : d is a metric distance. Let $\theta_{x,y}$ be the angle between x and y: $\theta_{x,y} = \arccos\frac{x\cdot y}{\|x\|\|y\|}$ (we denote $acos$ in the proof). We denote $\sin^+(x)$ the function such that $\sin^+(x) = \sin(x)$ on $[0,\pi/2]$ and $\sin^+(x) = 1$ on $[\pi/2,\pi]$. We can remark that $\sin^+(acos(c)) = \sqrt{1-c^2} = g(c)$ for $c \in [0,1]$. Finally, as time series have finite support, $\max_{o\in\mathcal{Z}} x\cdot\tau_o(y)$ is always non negative. We use now the fact that the angular distance is metric:

$$\begin{aligned}
\theta_{\tau_o(x),\tau_{o'}(y)} &\leq \theta_{\tau_o(x),z} + \theta_{z,\tau_{o'}(y)} \\
\min_{o,o'} \theta_{\tau_o(x),\tau_{o'}(y)} &\leq \min_{o} \theta_{\tau_o(x),z} + \min_{o'} \theta_{z,\tau_{o'}(y)} \\
\min_{o,o'} acos\frac{\tau_o(x)\cdot\tau_{o'}(y)}{\|x\|\|y\|} &\leq \min_{o} acos\frac{\tau_o(x)\cdot z}{\|x\|\|z\|} + \min_{o'} acos\frac{z\cdot\tau_{o'}(y)}{\|z\|\|y\|} \\
acos\frac{\max_{o,o'}\tau_o(x)\cdot\tau_{o'}(y)}{\|x\|\|y\|} &\leq acos\frac{\max_{o}\tau_o(x)\cdot z}{\|x\|\|z\|} + acos\frac{\max_{o'} z\cdot\tau_{o'}(y)}{\|z\|\|y\|}
\end{aligned}$$

This last inequality is obtained because arccos is non increasing. The next inequality uses the fact that $\sin^+$ is non decreasing and subadditive (see [4]).

$$\begin{aligned}
\sin^+\left(acos\frac{\max_{o,o'}\tau_o(x)\cdot\tau_{o'}(y)}{\|x\|\|y\|}\right) &\leq \sin^+\left(acos\frac{\max_o\tau_o(x)\cdot z}{\|x\|\|z\|}\right) \\
&\quad + \sin^+\left(acos\frac{\max_{o'} z\cdot\tau_{o'}(y)}{\|z\|\|y\|}\right) \\
g\left(\frac{\max_o x\cdot\tau_o(y)}{\|x\|\|y\|}\right) &\leq g\left(\frac{\max_o x\cdot\tau_o(z)}{\|x\|\|z\|}\right) + g\left(\frac{\max_o z\cdot\tau_o(y)}{\|z\|\|y\|}\right) \\
d(x,y) &\leq d(x,z)+d(z,y)
\end{aligned}$$

We consider now N time series $\{x_i\}_{1\leq i\leq N}$. We denote L the average length of these time series. Let K be the number of clusters, denoted C_k, and μ_k be the barycenter of cluster C_k. As explained in the introduction, EKSC, which uses d as dissimilarity measure, is a variant of the well-known K-means clustering model. It seeks to minimize the within-cluster sum of squares:

$$WCSS = \sum_{k=1}^{K}\sum_{x_i\in C_k} d^2(\mu_k, x_i)$$

EKSC proceeds by alternating between two steps: the assignment step which produces a new partition thanks to the metric distance d, and the update step which computes the new centroids (cluster barycenters). Contrary to the Euclidean case, computing a cluster barycenter is more technical when d (or d_+) is used as dissimilarity measure. Barycenter μ_k is obtained by minimizing $\mu_k = \arg\min_\mu \sum_{x_i\in C_k} d^2(\mu, x_i)$. Yang and Leskovec [1] assume that the optimal time shifts o_i have already been found during the previous assignment step. This allows for an exact extraction of μ_k thanks to the minimization of the following Rayleigh quotient: $\mu_k = \arg\min_\mu \frac{\mu^t T_k \mu}{\|\mu\|^2}$ with $T_k = \sum_{x_i\in C_k}\left(Id - \frac{\tau_i\ \tau_i^t}{\|\tau_i\|^2}\right)$, where $\tau_i = \tau_{o_i}(x_i)$. μ_k is therefore an eigenvector associated to the smallest eigenvalue of matrix T_k. In [1], a complete diagonalization of T_k is carried out to obtain the desired eigenvector. This approach is very costly (cubic complexity in L).

Two remarks can be made concerning the barycenter extraction procedure. We recall that if μ is an eigenvector, $-\mu$ is also an eigenvector. In the case of non negative time series, we can prove that the barycenter can be chosen as non negative. Moreover, in the case of arbitrary time series, the extraction procedure provides μ and $-\mu$ as solutions. But only one makes sense with respect to dissimilarity d. In order to choose between μ and its reverse counterpart, we memorize in each cluster the closest time series x_i to the barycenter. x_i is some kind of medoid. If $d(\mu_k, x_i) \leq d(-\mu_k, x_i)$, we choose μ_k as the barycenter, otherwise we choose $-\mu_k$.

1.3 Fast Implementation

Yang and Leskovec [1] point out the scalability issue of the KSC algorithm due mainly to the cubic complexity of the update step. In this section, we

propose four exact optimizations of EKSC (these optimizations can be applied to the original KSC as well). Thanks to these optimizations, the running time complexity of EKSC is reduced, which leads to large acceleration factors.

Computation of measure d thanks to Fourier transforms: As explained in [3], we can avoid testing all possible time shifts during the evaluation of measure d (this brute force approach has quadratic complexity in L). Indeed, we seek to maximize the cross-correlation function: $CC(o) = x \cdot \tau_o(y)$. We denote x^0 the vector x right padded with zeros. We denote ${}^0y$, the vector y left padded with zeros. We set $L(x^0) = L({}^0y) = L(x) + L(y) - 1$. It is well known that $CC \equiv \mathcal{F}^{-1}\left(\mathcal{F}(x^0)\,\mathcal{F}^*({}^0y)\right)$, where $\mathcal{F}$ (resp. $\mathcal{F}^{-1}$) is the Fourier transform (resp. the inverse Fourier transform), and * is the conjugate operator. Each evaluation of measure d therefore involves three Fourier transforms. With a fast algorithm (for instance, in the specific case of padded vectors, for which lengths are power of 2), the running time complexity is reduced from $O(L^2)$ to $O(L\ln(L))$. One interesting element which doesn't appear in [3] is that even if time series haven't got the same length, it is possible to precompute spectra once and for all before running the EKSC algorithm (initialization step). Thanks to this precomputation, during EKSC iterations, the evaluation of measure d involves only the computation of the inverse transform $\mathcal{F}^{-1}$. This remark leads to quite an important speed up.

Elkan's algorithm[2]: This algorithm, initially devised for K-Means, avoids a lot of measure evaluations thanks to the triangle inequality. It uses two cut off strategies. Let $\mu(x)$ be the currently assigned barycenter to the time series x. The first strategy is based on the fact that if another barycenter μ is located far enough from $\mu(x)$, then x can't be reassigned to μ. And therefore, the evaluation of $d(\mu, x)$ is avoided. The second strategy leverages the temporal continuity of the K-Means process: if the new position of the barycenter is close enough to the previous one, there is no reason for a time series to leave its current cluster. Once again, distance evaluations are avoided.

Power method: Firstly we can remark that the problem of a barycenter extraction is equivalent to this new maximization problem: $\mu_k = \arg\max_\mu \frac{\mu^t S_k \mu}{\|\mu\|^2}$, where $S_k = \sum_{x_i \in C_k} \frac{\tau_i\, \tau_i^t}{\|\tau_i\|^2}$. The eigenvector associated to the largest eigenvalue of this Rayleigh problem is the solution. We can prove that all eigenvalues are non negative, and therefore, we can use the power method which extracts the eigenvalue of largest magnitude. Therefore complete diagonalization of matrices S_k can be avoided, which is very efficient (diagonalization has cubic complexity). The power method is an iterative algorithm, which starts with an estimate of the desired eigenvector, and which produces a better approximation of the solution after each iteration. Each iteration simply involves a matrix vector product: $S_k\mu$. As a barycenter tends to move slowly during the end of the clustering process, the power method only has to carry out a few iterations to converge towards the updated barycenter.

Incremental computation of matrices S_k: At each iteration of EKSC, computing the K matrices S_k from scratch is time consuming. When the number of cluster modifications or shift modifications is low, it is more efficient to store

the matrices S_k in the memory, and to maintain them thanks to an incremental strategy. If the shift o_i or the assigned cluster of x_i is modified during the assignment step, the previous contribution $\frac{\tau_i \tau_i^t}{\|\tau_i\|^2}$ is removed from the previous matrix and the new contribution is added to the new matrix. In the case of a shift modification, the previous and the new matrices are the same ones. A broad rule to decide which of both computation strategies (incremental or from scratch) has to be prefered at each EKSC iteration can be easily devised: for instance we can compute the number of addition operations implied by both methods.

1.4 Centering time series

For the original KSC, the question of data centering was not addressed in [1]. Indeed, time series were assumed to be non negative, and therefore a representative barycenter has to be non negative. In this work, as we extend KSC to arbitrary time series, we may want to apply centering to the data. And moreover, we may wish to obtain centered barycenters. We follow the same path as in [3]. We consider the centering matrix $Q = Id - \frac{1}{L}\mathbf{1}$. Id is the identity matrix and $\mathbf{1}$ is the square matrix with only 1 as values. Q is symmetric and idempotent ($Q^2 = Q$). We maximize the Rayleigh quotient under the constraint $Q\mu = \mu$. We have $\mu_k = \arg\max_\mu \frac{\mu^t Q^t S_k Q \mu}{\|\mu\|^2}$. This time, the solution is obtained as an eigenvector associated to the largest eigenvalue of matrix $Q^t S_k Q$. As $Q^t S_k Q$ is symmetric and positive semidefinite, the power method can be used again (the largest eigenvalue is the largest eigenvalue in magnitude). Moreover, the incremental computation of matrices S_k can also be conserved if we center μ and $S_k\mu^c$ at each iteration of the power method (μ^c is μ after centering). In this case, no product with matrix Q has to be computed.

We show now that the solution μ_k is centered. We have $Q^t S_k Q \mu_k = \lambda_{max}\mu_k$. By multiplying by Q, we obtain $QQ^t S_k Q\mu_k = \lambda_{max} Q\mu_k$. And as $QQ^t S_k Q\mu_k = Q^t S Q\mu_k$, we obtain $\lambda_{max}\mu_k = \lambda_{max} Q\mu_k$. This implies $\mu_k = Q\mu_k$: μ_k is centered.

2 Experiments

In these experiments, we compare EKSC to a similar clustering model: K-shape [3]. K-Shape (KS) is based on a distinct dissimilarity measure $d_{KS}(x, y) = 1 - \frac{\max_o x \cdot \tau_o(y)}{\|x\|\|y\|}$. EKSC and K-Shape share the same update step. For EKSC, we present acceleration factors obtained thanks to the four optimizations. These optimizations are applied identically to K-Shape apart from Elkan's cut off. Indeed, the K-Shape dissimilarity d_{KS} is not metric [4].

We use five datasets, each with a ground truth partition (see the UCR archive `www.cs.ucr.edu/~eamonn/time_series_data`): ECG5000 (E), WordSynonyms (S), Fish (F), Non-Invasive Fetal ECG Thorax1 (N), Haptics (H). Table 1 reports dataset information. In all cases, the centered EKSC and KS are carried out ten times with random initializations. Parameter K is set to the number of classes of the ground truth partition. The stopping criterion is a threshold on

the relative inertia gain. As EKSC and KS don't have the same dissimilarity measure, inertia can't be compared. Therefore, we evaluate both models thanks to the Rand-Index measure (ground truth vs produced partitions). RI results are quite similar for both models, but thanks to Elkan, EKSC is significantly faster. Table 2 reports the acceleration factors for the three steps of EKSC: assignment, matrix computation, barycenter extraction. For the fourth dataset, the fully optimized EKSC is 405 times faster than EKSC with no optimization.

Data				EKSC					KS		
	Size	Len	Cl	RI	It	Elkan	~~Elkan~~	Base	RI	It	OKS
E	5 000	140	5	0.720	156	8	19	171	0.669	168	21
S	905	270	25	0.946	184	13	41	1 301	0.946	173	40
F	350	463	7	0.829	178	3	6	487	0.776	145	6
N	1 965	750	42	0.971	194	51	316	20 860	0.971	168	277
H	463	1 092	5	0.781	134	16	23	5 852	0.743	94	19

Table 1: Size: nb of time series - Len: time series length - Cl : nb of classes - RI: Rand-Index - It: cumulated nb of iterations (10 runs) - Elkan: optimized EKSC - ~~Elkan~~ : optimized EKSC with Elkan disabled - Base: EKSC with no optimization - OKS : optimized KS - Elkan, ~~Elkan~~, Base, OKS : cumulated running times in seconds

	$\frac{\text{Base}}{Elkan}$	$\frac{\text{~~Elkan~~}}{Elkan}$	ass. $\frac{\text{Base}}{Elkan}$	ass. $\frac{\text{~~Elkan~~}}{Elkan}$	mat. $\frac{\text{Base}}{Elkan}$	bar. $\frac{\text{Base}}{Elkan}$
E	21	2.3	19	2.7	5.1	112
S	101	3.2	53	4.5	4.0	258
F	159	1.9	82	3.3	6.7	358
N	405	6.1	388	11.0	7.4	678
H	364	1.4	96	2.4	4.8	1 029

Table 2: Quotient: acceleration factor compared to Base or ~~Elkan~~
Steps: assignment (ass.), matrix computation (mat.), barycenter extraction (bar.)

3 Conclusion

In this work, we present a natural extension of K-Spectral Centroid (EKSC) to arbitrary time series. We show that the new dissimilarity measure is a metric distance. Finally, we propose to speed up EKSC thanks to four exact optimizations, and show that this model is significantly faster than K-Shape.

References

[1] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proc. of the fourth ACM international conf. on Web search and data mining*, page 177. ACM, 2011.

[2] C. Elkan. Using the triangle inequality to accelerate k-means. In *Proc. of the Twentieth International Conference on Machine Learning*, ICML, page 147. The AAAI Press, 2003.

[3] John Paparrizos and Luis Gravano. Fast and accurate time-series clustering. *ACM Trans. Database Syst.*, 42(2):8:1–8:49, June 2017.

[4] S. Dongen and A. J. Enright. Metric distances derived from cosine similarity and pearson and spearman correlations. *CoRR*, abs/1208.3145, 2012.